



WebGrabber™

EULA, Copyright Notice & Trademarks

End User License Agreement

NOTICE TO USER: THIS IS A CONTRACT. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. This activePDF, Inc. ("ACTIVEPDF") End User License Agreement ("EULA") accompanies all activePDF products and related explanatory materials ("SOFTWARE"). The term "SOFTWARE" shall also include any upgrades, modified versions or updates of the Software licensed to you by ACTIVEPDF. Please read this EULA carefully. If you are reading this prior to electronic distribution, you will be asked to accept this agreement and continue to install or, if you wish to decline this agreement, in which case you will not be able to use the Software. The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold. **1. DEFINITIONS** I. SERVER - A single computer wholly owned, rented or leased by a single individual or entity on which one or more applications load and execute SOFTWARE in the memory space of that computer so that one or more users may access it. II. VIRTUAL SERVER - A single computer wholly owned, rented or leased by one individual or entity and who, in turn, rents or leases access to this computer to other individuals or entities and on which one or more applications load and execute SOFTWARE in the memory space of that computer so that multiple users may access it. III. DEVELOPMENT - The act of programming a tool or application that interacts with SOFTWARE. IV. APPLICATION SERVICE PROVIDER - An individual or entity ("OWNER") and who, in turn, charges other individuals or entities a fee for access to software applications wholly owned or licensed and maintained by the OWNER. The OWNER's application(s) load and execute SOFTWARE in the memory space of that computer so that multiple users may access it. **2. GRANT OF LICENSE** : Upon issuance of a valid SERIAL NUMBER (for SERVER and VIRTUAL SERVER LICENSES) by ACTIVEPDF to you, this EULA grants you the following rights: a. SOFTWARE PRODUCT : ACTIVEPDF grants to you as an individual, a non-transferable, nonexclusive license to use the SOFTWARE PRODUCT as follows: I. IF YOU PURCHASED A SERVER LICENSE - Usage is restricted to a single SERVER for a single individual or entity. All SERVERS that use this product must be licensed. III. IF YOU PURCHASED A VIRTUAL SERVER LICENSE - Usage is restricted to a single individual or entity. All other individuals or entities that can access or use SOFTWARE must purchase a separate VIRTUAL SERVER LICENSE. II. If you Purchased a Development License or a Developer Bundle. (SALES AND GRANT OF DEVELOPER LICENSES ARE AT ACTIVEPDF'S DISCRETION AND ALWAYS REQUIRE THE PURCHASE OR EXISTENCE OF VALID PRODUCTION LICENSES) You are hereby granted the right for development use only for the license or licenses purchased as development licenses. You must purchase a license for each developer granted access and for each development machine where you load the software. Development software is for Development purposes only and you may not use Development licenses in a live test or production environment (Test Servers require full licenses not Development Licenses). You may not resell, transfer, rent, or lease the SOFTWARE. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble, or otherwise attempt to discover the source code of the Software. You may not alter or modify in any way the installer for the Software, or create a new installer for the Software. You also agree to hold in the strictest confidentiality any and all code given to you by ACTIVEPDF to enable your application. b. If you are an entity, ACTIVEPDF grants you the right the same rights as above. c. UNTIL A VALID SERIAL NUMBER HAS BEEN ISSUED TO YOU BY ACTIVEPDF, YOU MAY ONLY USE THE SOFTWARE FOR A TRIAL PERIOD NOT TO EXCEED FIFTEEN (15) DAYS FROM THE INITIAL INSTALLATION. YOU AGREE TO REMOVE ANY COPIES OF THE SOFTWARE UPON EXPIRATION OF THE TRIAL PERIOD IF YOU DO NOT PURCHASE THE SOFTWARE. YOU MAY NOT DISTRIBUTE ANY PORTION OF THE SOFTWARE, INCLUDING THOSE PORTIONS OUTLINED UNDER "REDISTRIBUTABLE CODE" BELOW, UNTIL YOU ARE ISSUED A VALID SERIAL NUMBER. NO LICENSE IS GRANTED UNTIL THAT TIME. d. ELECTRONIC DOCUMENTS : Solely with respect to electronic documents included with the SOFTWARE, you may make an unlimited number of copies (either in hardcopy or electronic form), provided that such copies shall be used only for internal purposes and are not republished or distributed to any third party. e. STORAGE/NETWORK USE : You may store or install one (1) copy of the SOFTWARE on a storage device, such as a network server, for backup and archival purposes only. A license for the SOFTWARE may not be shared or used concurrently on different computers. f. You agree not to modify the Producer or Creator fields within any PDF documents created by SOFTWARE. g. ACTIVEPDF reserves all rights not expressly granted. **3. COPYRIGHT.** The SOFTWARE is owned by ACTIVEPDF and its suppliers, and its structure, organization and code are the valuable trade secrets of ACTIVEPDF and its suppliers. The SOFTWARE is protected by United States Copyright Law and International Treaty provisions. You may use trademarks only insofar as required to comply with Section 1 of this EULA and to identify printed output produced by the Software, in accordance with accepted trademark practice, including identification of trademark owner's name. Such use of any trademark does not give you any rights of ownership in that trademark. Except as stated above, this EULA does not grant you any intellectual property rights in the Software. **4. RESTRICTIONS.** You may not resell, transfer, rent or lease the SOFTWARE. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software. You may not alter or modify in any way the installer for the Software, or create a new installer for the Software. Redistribution rights, if any, are outlined below. **5. ADDITIONAL RESTRICTIONS.** I. For activePDF Server, activePDF DocConverter or activePDF WebGrabber : You agree that you will only use SOFTWARE on a SERVER and all applications that will access SOFTWARE will reside on the SERVER and you will not permit remote access to SOFTWARE except through your application residing on the SERVER. Additionally, you agree not to use SOFTWARE in a generic, bureau-like fashion whereby the only functionality of your application is to submit files for conversion to PDF. II. For

activePDF Server : You agree to not attempt to violate the internal licensing mechanism for processing thread increases. You also agree to not attempt to defeat said internal licensing mechanism by, directly or indirectly, halting and restarting the SOFTWARE. III. For activePDF DocConverter : You agree that you will not charge any fees, directly or indirectly, internally or externally, for access to SOFTWARE. This restriction includes charging per document, for the right to convert documents on your server and/or as a requirement of any sort. You also agree that should you violate this restriction, you will pay ACTIVEPDF an amount commensurate to the total amount collected by you for said restricted services or \$2,000 US, whichever is more. You also agree to surrender your license(s) immediately upon notification of violation of this restriction and that you will not be refunded any monies upon surrendering your license. You agree to not attempt to violate the internal licensing mechanism for document processing increases. You also agree to not attempt to defeat said internal licensing mechanism by, directly or indirectly, halting and restarting the SOFTWARE. IV. For activePDF WebGrabber : You agree to only render HTML to which you have a legal and lawful right to access and that you will not violate any individual's or entity's copyright protection with unauthorized access to that individual's or entity's content. You also agree to indemnify and hold harmless ACTIVEPDF from any such activity even if such violation is unintentional. V. For activePDF DocConverter : Installation and use by an APPLICATION SERVICE PROVIDER requires a separate, signed agreement and is prohibited by this EULA. **6. UPGRADES.** If the SOFTWARE is labeled as an upgrade, you must be properly licensed to use a product identified by ACTIVEPDF as being eligible for the upgrade in order to use the SOFTWARE. SOFTWARE labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this EULA. If the SOFTWARE is an upgrade of a component of a package of software programs that you licensed as a single product, the SOFTWARE may be used and transferred only as part of that single product package and may not be separated for use on more than one computer. Development licenses are not eligible for upgrades or other rights granted to products purchased with subscription. **7. REDISTRIBUTABLE CODE :** I. SERVER LICENSE - No portion of SOFTWARE PRODUCTS may be redistributed. II. VIRTUAL SERVER LICENSE - No portion of SOFTWARE PRODUCTS may be redistributed. **8. EXPORT RESTRICTIONS.** You agree that neither you nor your customers intend to or will, directly or indirectly, export or transmit (a) the SOFTWARE or related documentation and technical data or (b) your software products as defined under of this EULA (or any part thereof), or any process or service that is the direct product of the SOFTWARE to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. **9. NO WARRANTY.** The SOFTWARE is being delivered to you AS IS and ACTIVEPDF makes no warranty as to its use or performance. ACTIVEPDF AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. ACTIVEPDF AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL ACTIVEPDF OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF AN ACTIVEPDF REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. **10. GOVERNING LAW AND GENERAL PROVISIONS.** This EULA will be governed by the laws of the State of California, U.S.A., excluding the application of its conflicts of law rules. This EULA will not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If any part of this EULA is found void and unenforceable, it will not affect the validity of the balance of the EULA, which shall remain valid and enforceable according to its terms. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations. This EULA shall automatically terminate upon failure by you to comply with its terms. This Agreement may only be modified in writing signed by an authorized officer of ACTIVEPDF. **11. Notice to Government End Users.** If this product is acquired under the terms of a: GSA contract- Use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract; U.S. DoD contract- Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of 252.227-7013; Civilian agency contract- Use, reproduction, or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in this EULA. Unpublished-rights reserved under the copyright laws of the United States. activePDF, Inc., a California Corporation, 27405 Puerta Real, Suite 100, Mission Viejo, California 92691. Effective 3/1/2005

Copyright Notice

Copyright © 2000-2006, activePDF, Inc., All Rights Reserved Portions of this publication copyright © 2000 Gance AG. All Right Reserved. Released 02/2005 by activePDF, Inc. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from activePDF, Inc. 27405 Puerta Real, Suite 100, Mission Viejo, CA 92691-6314. **ALL EXAMPLES OF NAMES OR COMPANIES APPEARING IN THIS MANUAL ARE FICTIONAL. ANY SIMILARITY TO ACTUAL NAMES OR COMPANIES IS PURELY COINCIDENTAL.** Every effort has been made to ensure accuracy of this manual. However, activePDF, Inc. makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. activePDF, Inc. shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance or use of this manual or the examples herein. The information in this document is subject to change without notice.

Trademarks

activePDF[®], the activePDF Logo, activePDF Server[™], and activePDF DocConverter[™] are all registered trademarks or trademarks of activePDF, Inc. respectively. Adobe[®], Adobe Acrobat[®] and Adobe Acrobat Reader[®] are registered trademarks of Adobe[®] Systems, Inc. Microsoft[®], Microsoft Windows[®], Windows NT[®] and Windows 2000[®] are registered trademarks of Microsoft Corporation. All product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

Contacting activePDF, Inc.

- **activePDF, Inc. Corporate Headquarters**
- **Technical Support**
- **Sales**

activePDF, Inc. Corporate Headquarters

activePDF is a leading provider of PDF generation and manipulation tools. Our products are easy to use, enabling you to create PDF files that are compatible with 3rd party PDF tools and conform to the latest version of the PDF Specification, published by Adobe® Systems. **activePDF, Inc.** 27405 Puerto Real Suite 100 Mission Viejo, California 92691-6314 United States *Toll Free (U.S.): (866) GOTO PDF Elsewhere: +1-949-582-9002 Fax: (949) 582-9004 World Wide Web:*
www.activepdf.com

Technical Support

At activePDF, we are committed to providing you with timely answers to all of your technical support questions. For current hours of operation and details about support offerings, please visit the activePDF website at <http://www.activepdf.com/support/supportprogram/supportservices/>. activePDF's support services are subject to the terms and conditions in place at the time the service is used.

Sales

activePDF is a leading provider of server-side PDF generation and manipulation tools. For information on other activePDF product offerings, please contact activePDF's Sales Department by calling (866) GOTO PDF in the U.S., or +1-949-582-9002 elsewhere. You can also visit us on the web at www.activepdf.com.

Table of Contents

About this Document	12
Getting Started	13
System Requirements	14
Installation	15
activePDF License Administrator	16
Installing the Software	18
Configure the WGIELogin User	19
Upgrading WebGrabber's Version or Edition	20
From a Previous Version or Service Pack	21
From an Evaluation Edition	22
Testing your Installation	23
Native Engine Test	24
IE Engine Test	26
Object Reference	28
Instantiate the Object	29
Choosing the Rendering Engine	30
Native Engine	31
Methods	32
AddProxyExclusions	33
FontAddress	34
FontBlockQuote	35
FontDir	36
FontExample	37
FontHeading1	38
FontHeading2	39
FontHeading3	40
FontHeading4	41
FontHeading5	42
FontHeading6	43
FontListing	44
FontMenu	45
FontNormal	46
FontPreformatted	47
MainFont	48
SetPageNbrParams	49
Properties	50
AnchorColor	51
AuthenticName	52
AuthenticPassword	53
BrowserName	54
ConversationID	55
CookieControl	56
CookiesDir	57
FitToPage	58
FooterHeight	59
HeaderHeight	60
HTTPHeaders	61
IgnoreBaseInFile	62
ImageCacheKB	63
JPEGAllOnOnePage	64

LayoutHandling	65
LeftRightMargin	66
LoadImages	67
MaxSockets	68
PageBackColor	69
PageForeColor	70
PagesToCache	71
PaletteForceBackground	72
PrintBackgroundColors	73
PrintPageBreakForImages	74
PrintPageBreakForTables	75
ProxyPortHTTP	76
ProxyServerHTTP	77
ProxyServerPassword	78
ProxyServerUserName	79
Timeout	80
TopBottomMargin	81
UseCachedCopy	82
IE Engine	83
Properties	84
IEBottomMargin	85
IEDomain	86
IEFooter	87
IEHeader	88
IELeftMargin	89
IEOrientation	90
IEPassword	91
IEPrintFrames	92
IERightMargin	93
IETopMargin	94
IEUserName	95
IEWaitBeforePrinting	96
Common to Both Engines	97
Methods	98
AboutBox	99
CFGetPrt2DiskSettings	100
CleanUp	101
ClearFooterHTML	102
ClearHTMLText	103
ClearHeaderHTML	104
ClearVariables	105
DoPrint	106
GetStatus	107
Wait	109
Properties	111
CreateFromHTMLText	112
EMailAllow8Bits	113
EMailCC	114
EMailIDFrom	115
EMailNameFrom	116
EMailReplyTo	117
EMailSubject	118
EMailText	119
EMailTo	120
EngineToUse	121

FooterFirstPageOnly	122
FooterHTML	123
FooterStatic	124
FooterURL	125
FTPDirectory	126
FTPHost	127
FTPPassword	128
FTPUserName	129
HeaderFirstPageOnly	130
HeaderHTML	131
HeaderStatic	132
HeaderURL	133
HTMLTextToFile	134
HTTPPostData	135
LastError	136
PrinterName	137
PrinterTimeout	138
Prt2DiskSettings	139
SMTPServer	140
URL	141
UseFTP	142
Advanced Configuration	143
Using WebGrabber in .NET	144
Remote Operation	146
Preserving Background Colors, Images, and Table Borders	147
Using the Registry Switches	149
Appendices	151
PDF Coordinates and Units	152
Native Engine Supported HTML Tags	153
WebGrabber 3.5.2 Updates	155

About this Document

This document provides information for deploying activePDF WebGrabber in a development environment. It is organized into the following sections:

- ▶ **Getting Started** - Covers the System Requirements, Installation, Configuring the WGIELogin User, and Testing your Installation with each Engine.
- ▶ **Object Reference** - Includes object instantiation instructions and lists all methods and properties.
- ▶ **Advanced Configuration** - Covers the use of WebGrabber in .NET, Remote Operations, details Page Breaks, preserving Background Colors, and Registry Switches.
- ▶ **Appendices** - Provides an explanation of PDF Units, Native Engine Supported HTML Tags, and the Update History.

Who Should Read This Guide

This guide has been written for the developer who wants to programmatically generate PDFs and control the resultant output. The guide assumes you have a general knowledge of PostScript® and PDF, and that you are comfortable programming in a COM-enabled environment.

Documentation Feedback

activePDF strives to produce quality technical documentation. If you have comments or suggestions regarding our help files, PDF, print or online manuals please send an email to the [activePDF Documentation Team](#). Please include the following information in your message:

- ▶ Product name and version number
- ▶ Print manual, PDF or help file
- ▶ Section or Topic title
- ▶ Brief description of content
- ▶ Your suggestion for improvement or correction

NOTE: This email address is only for documentation feedback. If you have a technical question, please contact [Technical Support](#).

Getting Started

activePDF WebGrabber allows you to dynamically convert HTML to PDF - any URL, raw HTML, or HTML file to PDF on the fly - while maintaining embedded styles. WebGrabber also affords you full control over headers, footers and page breaks, making it easy to turn your existing web application into a PDF reporting engine. With a few lines of code, you can begin converting virtually anything viewed in your browser to PDF.

- › [System Requirements](#)
- › [Installation](#)
- › [Testing your Installation](#)

System Requirements

Operating System Requirements

- ▶ Microsoft® Windows NT® 4.0 (Service Pack 5 minimum) or
- ▶ Microsoft Windows® 2000 or
- ▶ Microsoft Windows Server 2003

Minimum Recommended Hardware Requirements

- ▶ Pentium III 500-MHz or higher
- ▶ 128 MB of RAM
- ▶ 15MB of Hard Disk Space (for application)

Port Requirements

- ▶ By default, port 64320 must remain open and dedicated for WebGrabber.

Additional Software Requirements

- ▶ activePDF Server™
- ▶ Internet Explorer 5.5 or higher if you are using the [IE Engine](#).

Installation

Installing the registered production edition of activePDF WebGrabber requires a *Product Serial Number*, the [activePDF License Manager](#), and the installation software. The *Product Serial Number* is supplied via email upon purchase. If you are installing the evaluation edition, proceed directly to [Installing the Software](#).

- ▶ [activePDF License Manager](#)
- ▶ [Installing the Software](#)
- ▶ [Configure the WGIELogin User](#)
- ▶ [Upgrading WebGrabber's Version or Edition](#)

```
</PLAINTEXT><PRE></PRE></Q></S></SAMP></SCRIPT></SELECT></SMALL></STRIKE></STRONG></SUB></
<UL></UL></VAR></WBR><XMP></XMP><FONT face=arial></FONT>
<SCRIPT language=JavaScript>
    function showHide(targetName) {
        if( document.getElementById ) { // NS6+
            target = document.getElementById(targetName);
        } else if( document.all ) { // IE4+
            target = document.all[targetName];
        }

        if( target ) {
            if( target.style.display == "none" ) {
                target.style.display = "inline";
            } else {
                target.style.display = "none";
            }
        }
    }
</SCRIPT>
```

```
<TABLE borderColor=#000808 cellPadding=3 bgColor=#e7e7e7 border=1>
<TBODY>
<TR>
<TD bgColor=#000066><FONT style="FONT: 11pt/13pt verdana; COLOR: white"
    color=white>Error Occurred While Processing Request</FONT></TD></TR>
<TR>
<TD><FONT style="FONT: 8pt/11pt verdana; COLOR: black">
<TABLE cellSpacing=0 cellPadding=0 width=500 border=0>
<TBODY>
<TR>
<TD id=tableProps2 vAlign=center align=left width=500>
<H1 id=textSection1
    style="FONT: 13pt/15pt verdana; COLOR: black">Variable VARPRODUCT is
    undefined. </H1></TD></TR>
<TR>
<TD id=tablePropsWidth width=400 colSpan=2><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black"></FONT></TD></TR>
<TR>
<TD>&nbsp;</TD></TR>
<TR>
<TD colSpan=2><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Please try the
    following:
<UL>
<LI>Enable Robust Exception Information to provide greater detail
    about the source of errors. In the Administrator, click Debugging
    & Logging & Debugging Settings, and select the Robust
    Exception Information option.
<LI>Check the <A
    href="http://www.macromedia.com/go/proddoc_getdoc"
    target=new>ColdFusion documentation</A> to verify that you are
    using the correct syntax.
<LI>Search the <A
    href="http://www.macromedia.com/support/coldfusion/"
    target=new>Knowledge Base</A> to find a solution to your problem.
</LI></UL>
<P></P></FONT></TD></TR>
<TR>
<TD colSpan=2>
<TABLE cellSpacing=0 cellPadding=0 border=0>
<TBODY>
```

```
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Browser&nbsp;&nbsp;&nbsp;</FONT><
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Mozilla/4.0
    (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; .NET
    CLR 2.0.50727)</FONT></TD></TR>
<TR>
  <TD><FONT style="FONT: 8pt/11pt verdana; COLOR: black">Remote
    Address&nbsp;&nbsp;&nbsp;</FONT></TD>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">216.23.175.190</FONT></TD><
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Referrer&nbsp;&nbsp;&nbsp;</FONT>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black"></FONT></TD></TR>
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Date/Time&nbsp;&nbsp;&nbsp;</FONT>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">16-Oct-07 03:00
    PM</FONT></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></FONT></TD>
```

Installing the Software

Production and Evaluation Editions of activePDF WebGrabber are available for download from <http://www.activepdf.com/downloads/serverproducts/index.cfm>. activePDF WebGrabber requires activePDF Server™. Refer to the [activePDF Server Product Manual](#) for activePDF Server installation instructions. The activePDF WebGrabber Service must run under the *Local System* account. If Internet Explorer 5.5 or higher is detected, you will be prompted to [Configure the WGIELogin User](#) upon completing the installation. The account must have *Administration* privileges on the local machine.

1. Log in as the **Local Administrator**
2. Insert your activePDF CD or download WebGrabber from <http://www.activepdf.com/downloads/serverproducts/index.cfm>.

NOTE: If you have purchased, complete the steps for the [License Manager](#) at this time.

3. Double-click **APWebGrb.EXE** and then follow the onscreen instructions to complete the installation.

NOTE: If Internet Explorer 5.5 or higher is detected, you will be prompted to [Configure the WGIELogin User](#) at this time.

4. Restart your computer
-

Configure the WGIELogin User

The *WGIELogin* user account is required to use the **IE Engine**. During the WebGrabber process, the *WGIELogin* user instantiates Internet Explorer to grab and render the **URL** before printing to PostScript®. Normally, you are prompted to configure the *WGIELogin* user if Internet Explorer 5.0 or higher is detected upon completing the **Installation**.

NOTE: The *WGIELogin* user must have *Administrative* privileges on the local machine.

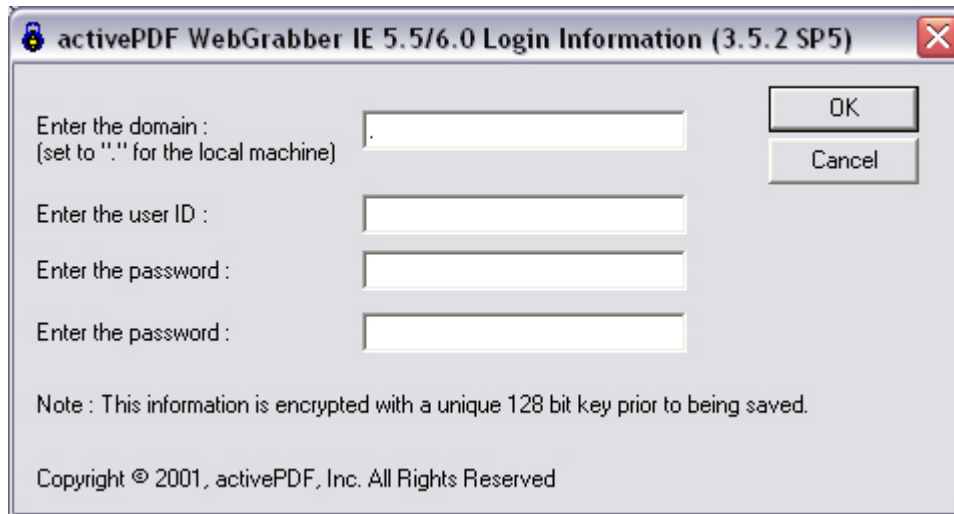


Figure 2.2: activePDF WebGrabber IE 5.5/6.0 Login Information Dialog.

1. Log on as the **Local Administrator**.
2. Click **Start > Run**, type the location of the *APWebGrabber* folder (By default, the location is `X:\Program Files\activePDF\APWebGrabber`), and then click **OK**
3. Double-click **WGIELogin.exe**
4. For the *domain*, type the **network domain** or **period** (.) for the local machine.
5. For the *UserID*, specify the **user account with Administrative privileges** to specify as the *WGIELogin* user.
6. For the *Password*, type the specified *UserID* **password**.
7. For the *Password*, re-type the specified *UserID* **password** for confirmation.
8. Click **OK**.
9. Click **Start > Run**, type `services.msc /s`, and then click **OK**.
10. Right-click the **Print Spooler** service, and then click **Restart**.

NOTE: Restarting the *Print Spooler* service will restart all system printing processes.

11. Click **Start > Run**, type `services.msc /s`, and then click **OK**.

NOTE: You must repeat these steps if the *WGIELogin* user changes for any reason.

Upgrading WebGrabber's Version or Edition

It is possible to upgrade from a previous service pack as well as from an evaluation edition without re-applying your *Product Serial Number*. This makes it easy to transition from testing to the latest enhancements.

- **From a Previous Version or Service Pack**
- **From an Evaluation Edition**

From a Previous Version or Service Pack

Periodically, activePDF releases Service Pack upgrades including updates and fixes for the current version of WebGrabber.

NOTE: You must install the corresponding version or edition of activePDF Server. In addition, you must update your **.NET Wrapper**.

1. Log in as the **Local Administrator**
 2. Download the latest service pack available from <http://www.activepdf.com/downloads/servicepacks/index.cfm>.
 3. Double-click the downloaded service pack **EXE** file, and then follow the onscreen instructions to complete the installation.
 4. Restart your computer
-

From an Evaluation Edition

The evaluation edition of activePDF WebGrabber is a fully functional and free to use. The evaluation period expires after 15-days and all output contains the activePDF brand. After purchasing a production license for activePDF WebGrabber, you can upgrade your evaluation edition by adding the serial number using the [activePDF License Manager](#). Simply add the number to remove the evaluation watermark.

NOTE: You may need to restart your server-machine before adding the serial number.

Testing your Installation

activePDF WebGrabber includes two different **Rendering Engines** for the process of converting HTML to PDF. You will need test to the **Native** and **IE** engines using separate scripts.

- ▶ **Native Engine Test**
- ▶ **IE Engine Test**

Native Engine Test

The following VBScript example should be used to test your WebGrabber installation with the **Native Engine**. Refer to the [Object Reference](#) for a detailed explanation of the object instantiation and usage requirements.

NOTE: Refer to the [IE Engine Test](#) for the **IE Engine**.

This file is required:

- ▶ **BasicNative.vbs** - Code detail provided below.

1. Log in as the **Local Administrator**.
2. Click here to download the example (*includes required files*):



3. Unzip the contents to your desktop.
4. Double-click **BasicNative.vbs** to run the script.

```
' Get local path as varPath
arrayScr = Split(WScript.ScriptFullName, "\", -1, 1)
For i = 0 to UBound(arrayScr) - 1
    varPath = varPath & arrayScr(i) & "\"
Next

' Define Variables
' EngineToUse (1=IE, 0=Native)
varEngine = 0

' URL format: "http://www.activepdf.com" or "file:///folder/file.htm"
varURL = "http://www.activepdf.com/support/knowledgebase/view.cfm?kb=10543&code=vbs"

' Timeout for various calls
varTimeout = 60

' For local machine use these defaults
varIP = "127.0.0.1"
varPort = "64320"

' Instantiate Objects
Set WG = CreateObject("APWebGrabber.Object")
Set APS = CreateObject("APServer.Object")

' Set activePDF Server properties
APS.OutputDirectory varPath
APS.PDFTimeout = varTimeout

' Must call before DoPrint to pass server settings to WG
WG.Prt2DiskSettings = APS.ToString()

' Set WebGrabber properties
WG.EngineToUse = varEngine
WG.URL = varURL
WG.TimeOut = varTimeout
WG.PrinterTimeout = varTimeout
```

```

WG.PrintBackgroundColors = 1
WG.TopBottomMargin = 150
WG.LeftRightMargin = 150

' Start the conversion process
varReturn = WG.DoPrint(varIP, varPort)
If varReturn <> 0 Then Error("DoPrint") End If

' Wait for conversion result
varReturn = WG.Wait(varIP, varPort, varTimeout, "")
If varReturn <> 19 Then Error("Wait") End If

' Run WG CleanUp
WG.Cleanup varIP, varPort

' Clear Objects
Set WG = Nothing
Set APS = Nothing

' Error Handling
Sub Error(Method)
    MsgBox "" & Method & "" failed with a "" & varReturn & _
        "" & VBCRLF & "KB article on Return codes:" & VBCRLF & _
        "http://www.activepdf.com/support/knowledgebase/viewKb.cfm?id=10033&tk=ts"
    Set WG = Nothing
    Set APS = Nothing
    Wscript.Quit
End Sub

```

IE Engine Test

The following VBScript example can be used to test your WebGrabber installation using the **IE Engine**. Refer to the [Object Reference](#) for a detailed explanation of the object instantiation and usage requirements.

NOTE: You must have Internet Explorer 5.5 or higher installed and configure the **WGIELogin User** before running the script. Refer to the [Native Engine Test](#) for the **Native Engine**.

This file is required:

- ▶ **BasicIE.vbs** - Code detail provided below.

1. Log in as the **Local Administrator**.
2. Click here to download the example (*includes required files*):



3. Unzip the contents to your desktop.
4. Double-click **BasicIE.vbs** to run the script.

```
' Get local path as varPath
arrayScr = Split(WScript.ScriptFullName, "\", -1, 1)
For i = 0 to UBound(arrayScr) - 1
    varPath = varPath & arrayScr(i) & "\"
Next

' Define Variables
' EngineToUse (1=IE, 0=Native)
varEngine = 1

' URL format: "http://www.activepdf.com" or "file://folder/file.htm"
varURL = "http://www.activepdf.com/support/knowledgebase/view.cfm?kb=10544&code=vbs"

' Timeout for various calls
varTimeout = 60

' For local machine use these defaults
varIP = "127.0.0.1"
varPort = "64320"

' Instantiate Objects
Set WG = CreateObject("APWebGrabber.Object")
Set APS = CreateObject("APServer.Object")

' Set activePDF Server properties
APS.OutputDirectory varPath
APS.PDFTimeout = varTimeout

' Must call before DoPrint to pass server settings to WG
WG.Prt2DiskSettings = APS.ToString()

' Set WebGrabber properties
WG.EngineToUse = varEngine
WG.URL = varURL
WG.TimeOut = varTimeout
```

```

WG.PrinterTimeout = varTimeout

' Start the conversion process
varReturn = WG.DoPrint(varIP, varPort)
If varReturn <> 0 Then Error("DoPrint") End If

' Wait for conversion result
varReturn = WG.Wait(varIP, varPort, varTimeout, "")
If varReturn <> 19 Then Error("Wait") End If

' Run WG CleanUp
WG.Cleanup varIP, varPort

' Clear Objects
Set WG = Nothing
Set APS = Nothing

' Error Handling
Sub Error(Method)
    MsgBox "" & Method & " failed with a " & varReturn & _
        "" & VBCRLF & "KB article on Return codes:" & VBCRLF & _
        "http://www.activepdf.com/support/knowledgebase/viewKb.cfm?id=10033&tk=ts"
    Set WG = Nothing
    Set APS = Nothing
    Wscript.Quit
End Sub

```

Object Reference

activePDF Server is controlled via a programmable COM object. This section contains the technical documentation for the [WebGrabber Engines](#) and the *Methods* and *Properties* contained in each. Once you have become familiar with [Instantiating the WebGrabber Object](#) and [Choosing the Rendering Engine](#), you can use this as your main resource for understanding WebGrabber.

- ▶ [Instantiate the Object](#)
- ▶ [Choosing the Rendering Engine](#)

Instantiate the Object

The WebGrabber COM object is represented by the *Class ID* `APWebGrabber.Object`. You can use this *Class ID* to instantiate the object.

Instantiation

The instantiation of the WebGrabber COM object using the *Class ID* depends on your scripting environment. We recommend the use of *Late-Binding* as the WebGrabber type library is already registered in the *Object Registry*. The following VBScript example demonstrates the instantiation process.

```
' Instantiate Object
Set objWG = CreateObject("APWebGrabber.Object")
```

Clean-Up

Certain scripting environments support clearing the variables. Destroying the object prevents the memory build-up after the conversion is finished. Below is a VBScript example that would appear at the end of your code.

```
' Release Object
Set objWG = Nothing
```

Choosing the Rendering Engine

activePDF WebGrabber uses the **Native** or **IE** rendering engine to render the specified **URL** in a process little different from an ordinary Browser. However, as with Browsers, each rendering engine renders the same **URL** in different ways. By understanding the abilities of each engine and their unique or common methods and properties, you can use the best rendering engine for your implementation. Use the **EngineToUse** property to specify the engine.

- **Native Engine**
- **IE Engine**
- **Common to Both Engines**

Native Engine

The *Native Engine* is the default rendering engine. It provides faster conversion speeds than the *IE Engine*, but is not as robust. Using [custom tag for page breaks](#) is supported. Refer to [Native Engine Supported HTML Tags](#) for a list of supported tags. Set the [EngineToUse](#) property equal to 0 (*zero*) to use the *Native Engine*. The *Native Engine* includes many [Methods](#) and [Properties](#) useful for controlling output font parameters and breaks.

- [Methods](#)
- [Properties](#)

Methods

The *Native Engine* methods are primarily designed to help you control font usage in your rendered output PDF, but also include useful Proxy and page number options.

- **AddProxyExclusions**
- **FontAddress**
- **FontBlockQuote**
- **FontDir**
- **FontExample**
- **FontHeading1**
- **FontHeading2**
- **FontHeading3**
- **FontHeading4**
- **FontHeading5**
- **FontHeading6**
- **FontListing**
- **FontMenu**
- **FontNormal**
- **FontPreformatted**
- **MainFont**
- **SetPageNbrParams**

AddProxyExclusions

Description: Defines the IP address of domains accessed directly regardless of the [ProxyServerHTTP](#) settings, which can be useful when conversion takes place internally and externally.

Syntax: `object.AddProxyExclusions "ExcludedDomain"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
ExcludedDomain	STRING	The domain address of the destination host.

Remarks: The exclusion list is tail-matched. For example, specifying host "x.y.com" would include "a.x.y.com".

FontAddress

Description: Sets the font used for any <ADDRESS> tags when rendering the URL to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontAddress "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontBlockQuote

Description: Sets the font used for any <BLOCKQUOTE> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontBlockQuote "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontDir

Description: Sets the font used for any <DIR> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontDir "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontExample

Description: Sets the font used for any <SAMP> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontExample "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading1

Description: Sets the font used for any <H1> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading1 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading2

Description: Sets the font used for any <H2> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading2 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading3

Description: Sets the font used for any <H3> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading3 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading4

Description: Sets the font used for any <H4> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading4 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading5

Description: Sets the font used for any <H5> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading5 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontHeading6

Description: Sets the font used for any <H6> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontHeading6 "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontListing

Description: Sets the font used for any <TT>, <KBD>, or <CODE> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontListing "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontMenu

Description: Sets the font used for any <MENU> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontMenu "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontNormal

Description: Sets the normal font used to render [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontNormal "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

FontPreformatted

Description: Sets the font used for any <PRE> tags when rendering the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.FontPreformatted "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

MainFont

Description: Sets the default font used to render the [URL](#) to PDF.

NOTE: Does not override explicit settings.

Syntax: `object.MainFont "Name", Size, Bold, Italic, Underline, Strikethrough`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	Name	STRING	The font name appearing in the X:\WINDOWS\Fonts or X:\WINNT\Fonts. NOTE: The font must be recognized by the activePDF Server Font Utility before it can be used.
	Size	INTEGER	The size of the font in PDF Units .
	Bold	BOOLEAN	Enables or disables the bold format. True = Enabled. False = Disabled. <i>(Default)</i>
	Italic	BOOLEAN	Enables or disables the italic format. True = Enabled. False = Disabled. <i>(Default)</i>
	Underline	BOOLEAN	Enables or disables the underline format. True = Enabled. False = Disabled. <i>(Default)</i>
	Strikethrough	BOOLEAN	Enables or disables the strikethrough format. True = Enabled. False = Disabled. <i>(Default)</i>

SetPageNbrParams

Description: Sets the starting and total page number passed with header and footer calls.

Syntax: `object.SetPageNbrParams (StartPageNbr, TotalPages)`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
StartPageNbr	INTEGER	If used, the starting page number passed in the header and footer. Set to 0 (zero) for the default value.
TotalPages	INTEGER	If used, the total page number passed in the header and footer. Set to 0 (zero) for the default value.

Properties

The *Native Engine* properties enable you to fine tune the look of your output PDF, but also include several unique properties, such as [JPEGAllOnOnePage](#).

- **AnchorColor**
- **AuthenticName**
- **AuthenticPassword**
- **BrowserName**
- **ConversationID**
- **CookieControl**
- **CookiesDir**
- **FitToPage**
- **FooterHeight**
- **HeaderHeight**
- **HTTPHeaders**
- **IgnoreBaseInFile**
- **ImageCacheKB**
- **JPEGAllOnOnePage**
- **LayoutHandling**
- **LeftRightMargin**
- **LoadImages**
- **MaxSockets**
- **PageBackColor**
- **PageForeColor**
- **PagesToCache**
- **PaletteForceBackground**
- **PrintBackgroundColors**
- **PrintPageBreakForImages**
- **PrintPageBreakForTables**
- **ProxyPortHTTP**
- **ProxyServerHTTP**
- **ProxyServerPassword**
- **ProxyServerUserName**
- **Timeout**
- **TopBottomMargin**
- **UseCachedCopy**

AnchorColor

Description: Sets the color used for hyperlink text in the output PDF.

NOTE: Does not override explicit settings.

Syntax: `object.AnchorColor = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	LONG	The default color for hyperlink text in hex for the output PDF. For example, &H00FFFFFF. Defaults to blue.

AuthenticName

Description: The user name passed for a [URL](#) requiring *Basic Authentication*.

Syntax: `object.AuthenticName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The user name for a URL requiring <i>Basic Authentication</i> .

AuthenticatePassword

Description: The password for the [AuthenticateName](#).

Syntax: `object.AuthenticatePassword = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The password for the AuthenticateName . Defaults to blank.

BrowserName

Description: Sets the browser name passed as the *HTTP User Agent header*.

Syntax: `object.BrowserName = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	The browser name passed as the <i>HTTP User Agent header</i> . For example, "Mozilla/2.0".

NOTE: Can be used to help avoid rejection by servers.

ConversationID

Description: Sets the Conversation ID used by WebGrabber component to communicate with the WebGrabber server-machine in order to retrieve status and port document names.

Syntax: `object.ConversationID = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	The Conversation ID used during WebGrabber component and WebGrabber server-machine communications. By default, a unique ID is created.

CookieControl

Description: Sets how cookies are handled during the current session.

NOTE: WebGrabber does not use cookies, but they are part of normal web protocol.

Syntax: `object.CookieControl = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	A value indicating how cookies are handled. 0 = Always submit cookies. <i>(Default)</i> 1 = Never submit cookies. 2 = Only submit cookies received in the current session.

CookiesDir

Description: Sets the location where cookies retrieved during the current session are stored.

NOTE: WebGrabber does not use cookies, but they will be retrieved as part of normal web protocol.

Syntax: `object.CookiesDir = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The full path to the directory to store HTTP cookies.

FitToPage

Description: Instructs WebGrabber to render the HTML to a single page.

NOTE: Output is printed using 72 DPI and **PaperSize** is ignored. To reduce the HTML to fit on a single page, use the activePDF Server™ **Scaling** property.

Syntax: `object.FitToPage = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	A value instructing WebGrabber render the input data to a single page. True = Input data rendered to a single page. False = Input data renders to multiple pages. <i>(Default)</i>

FooterHeight

Description: Defines the height of the document footer.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.FooterHeight = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	FLOAT	The height of the footer in inches.

Remarks: If **TopBottomMargin** is used, the height of the footer is measured from the value passed with that property. For example, if the margin is *1 in.* and the footer height is *1 in.*, the top of the footer starts *2 in.* from the bottom of the page and ends *1 in.* from the bottom.

HeaderHeight

Description: Defines the height of the document header.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.HeaderHeight = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	FLOAT	The height of the header in inches.

Remarks: If **TopBottomMargin** is used, the height of the header is measured from the value passed with that property. For example, if the margin is *1 in.* and the header height is *1 in.*, the top of the header starts *1 in.* from the top of the page and ends *2 in.* from the top.

HTTPHeaders

Description: Sets data passed to the [URL](#) as additional headers during conversion.

Syntax: `object.HTTPHeaders = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	Data passed as additional headers when rendering the URL to PDF. Headers must use the following format: "header: data" You can separate multiple headers using a <i>Carriage Return</i> and <i>Line Feed</i> combination.

IgnoreBaseInFile

Description: Instructs WebGrabber to ignore any <BASE> tags in the [URL](#).

Syntax: `object.IgnoreBaseInFile = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	A value instructing WebGrabber to ignore any <BASE> tag found in the input data. 0 = <BASE> tag not ignored. <i>(Default)</i> -1 = <BASE> tag is ignored.

ImageCacheKB

Description: Sets the maximum size assigned for storing images in the local cache.

Syntax: `object.ImageCacheKB = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	INTEGER	The maximum size in kilobytes (KB) assigned for storing images in the local cache. Oldest images are overwritten as needed. Defaults to 0 (zero).

NOTE: This property does not affect images stored due to the [PagesToCache](#) property. Cached images might consume resources.

JPEGAllOnOnePage

Description: Instructs WebGrabber to render the entire page JPEG instead of PDF.

Syntax: `object.JPEGAllOnOnePage = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	A value instructing WebGrabber to render the page to JPEG. True = The entire page renders to JPEG. False = The page renders to PDF per method and property calls. <i>(Default)</i>

LayoutHandling

Description: Sets when WebGrabber calculates the layout during conversion. By default, the text is loaded, the layout is calculated, and then the document is paginated for PDF output.

Syntax: `object.LayoutHandling = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	SHORT	<p>A value indicating how WebGrabber should calculate layout.</p> <p>0 = Layout calculates as HTTP text loads. <i>(Default)</i></p> <p>1 = Layout calculation delayed until the HTTP text is fully loaded.</p> <p>2 = Layout is not calculated.</p> <p>NOTE: Default is recommended. Simple layouts can benefit from disabling layout calculation.</p>

LeftRightMargin

Description: Sets the left and right page margins for the output.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.LeftRightMargin = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	LONG	The distance between the left and right edge of the printed page and the rendered data in pixels for the output PDF. Actual width is determined by the DPI of the DefaultPrinter . For example, 150 pixels equals ½ inch on a 300 DPI printer (<i>Pixels / DPI = Actual Width</i>). Defaults to 0 (zero).

LoadImages

Description: Determines how WebGrabber loads images.

Syntax: `object.LoadImages = setValue`

Arguments:

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	BOOLEAN	A value indicating how WebGrabber should load images. True = Images load as normal. <i>(Default)</i> False = Images are not loaded.

NOTE: Document does not need to be parsed for hyperlinks. Will not work if images are in the header or footer.

MaxSockets

Description: Sets the maximum number of sockets used during page retrieval, which can reduce network utilization and conserve resources.

Syntax: `object.MaxSockets = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	The maximum number of sockets used during page retrieval. Defaults to 2.

PageBackColor

Description: Sets the default color for the background of the [URL](#) during conversion.

NOTE: Does not override explicit settings.

Syntax: `object.PageBackColor = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	LONG	The color for the page background in hex. For example, &H00FFFFFF. Defaults to white.

PageForeColor

Description: Sets the default color for all text in foreground of the [URL](#) during conversion.

NOTE: Does not override explicit settings.

Syntax: `object.PageForeColor = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	LONG	The color for all text in hex. For example, &H00FFFFFF. Defaults to black.

PagesToCache

Description: Sets the number of pages stored in the local cache during the open session.

Syntax: `object.PagesToCache = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	INTEGER	A value indicating the number of pages to store in the local cache during the open session. Oldest pages are overwritten as needed. Defaults to 0 (zero).

NOTE: Cached pages might consume resources.

PaletteForceBackground

Description: Allows the container to impose a palette for WebGrabber to use during conversion to PDF.

NOTE: Typically, this will only work with forms containing multiple palette-aware objects.

Syntax: `object.PaletteForceBackground = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	A value instructing WebGrabber to post to the FTP. 0 = Do not use palette control. <i>(Default)</i> -1 = Use the palette control.

PrintBackgroundColors

Description: Instructs WebGrabber to print zero, table cell, or all background colors.

Syntax: `object.PrintBackgroundColors = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	INTEGER	<p>A value instructing WebGrabber to print zero, table cell, or all background colors.</p> <p>0 = Background colors will not print. <i>(Default)</i></p> <p>1 = Only print table cell background colors.</p> <p>2 = Only print table cell background colors.<i>(Must specify color printer)</i></p> <p>3 = Print all background colors.</p> <p>NOTE: Background prints as dark gray (#A8A8A8) if page does not specify background color.</p> <p>4 = Print all background colors.<i>(Must specify color printer)</i></p>

Remarks: The *Native Engine* prints all images.

PrintPageBreakForImages

Description: Defines the area in which if an image tag () appears, but the image does not fit, WebGrabber issues a page break and the image prints to the next page. WebGrabber splits any images starting above the region but not fitting in the region..

Syntax: `object.PrintPageBreakForImages = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	INTEGER	<p>A value indicating the percentage from the bottom of the page in which if the image tag () appears, but the image does not fit, WebGrabber issues a page break. WebGrabber splits any images starting above the region but not fitting in the region.</p> <p>For example, setting this value to 50 instructs WebGrabber to issue a page break before any image starting below but the middle, but not ending before the end of the page. WebGrabber splits any image that starts before the middle of the page but does not fit on the page.</p> <p>100 = WebGrabber adds a page break before all images.</p> <p>20 = WebGrabber adds a page break before any images starting below and not fitting in the bottom 20% of the page and splits any image starting before the bottom 20%, but not finishing. (<i>Default</i>)</p> <p>0 = WebGrabber splits any images not fitting on the page.</p>

PrintPageBreakForTables

Description: Defines the area from the bottom of the page in which if a table cell (<TR><TD></TD></TR>) does not fit, WebGrabber issues a page break and prints the table cell to the next page. WebGrabber splits any cells starting above the region but not fitting in the region.

Syntax: `object.PrintPageBreakForTables = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	INTEGER	<p>A value indicating the percentage from the bottom of the page in which if a table cell (<TR><TD></TD></TR>) starts, WebGrabber issues a page break. WebGrabber splits any cells starting above the region but not fitting in the region.</p> <p>For example, setting this value to 50 instructs WebGrabber to issue a page break before any table cell starting below the middle of the page. WebGrabber splits any cells starting before the middle of the page but not fitting in the lower half of the page.</p> <p>100 = WebGrabber adds a page break before all table cells not fitting on the page.</p> <p>20 = WebGrabber adds a page break before any table cells starting below the bottom 20% of the page and splits any table cell starting before the bottom 20%, but not fitting. (<i>Default</i>)</p> <p>0 = WebGrabber splits all table cells not fitting on the page.</p>

ProxyPortHTTP

Description: The port for the specified [ProxyServerHTTP](#).

Syntax: `object.ProxyPortHTTP = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The port for the specified ProxyServerHTTP . Defaults to 0 (zero).

ProxyServerHTTP

Description: The IP address of the proxy server used as a gateway to the remote host.

Syntax: `object.ProxyServerHTTP = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	The IP address of the proxy server used as a gateway to the remote host. Defaults to blank.

Remarks: Invalid proxy settings can result in an *Authenticate Header* message or a *407 HTTP response*.

ProxyServerPassword

Description: The password for the [ProxyServerUserName](#).

Syntax: `object.ProxyServerPassword = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The password for the ProxyServerUserName . Defaults to blank.

ProxyServerUserName

Description: The user name for the specified [ProxyServerHTTP](#).

Syntax: `object.ProxyServerUserName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The user name for the specified ProxyServerHTTP . Defaults to anonymous.

Timeout

Description: The time WebGrabber waits between locating an available thread and connecting to the [URL](#).

NOTE: Increasing this time can be useful when using large queries.

Syntax: `object.Timeout = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	The connection timeout in seconds. Defaults to 30.

TopBottomMargin

Description: Defines the size of the top and bottom margin.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.TopBottomMargin = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	LONG	The distance between the top and bottom edge of the printed page and the rendered data in pixels for the output PDF. Actual width is determined by the DPI of the DefaultPrinter . For example, 150 pixels equals ½ inch on a 300 DPI printer (<i>Pixels / DPI = Actual Width</i>). Defaults to 0 (zero).

UseCachedCopy

Description: Uses the cached copy of the specified [URL](#) if [PagesToCache](#) is enabled and a cached page exists.

Syntax: `object.UseCachedCopy = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	A value instructing WebGrabber to use the cached copy. 0 = Do not use cached copy. <i>(Default)</i> 1 = Use cached copy.

IE Engine

The *IE Engine* uses the [WGIELogin User](#) to instantiate Internet Explorer for rendering. It can provide truer quality output, but is not as fast as the [Native Engine](#). The *IE Engine* supports:

- ▶ Cascading Style Sheets (*CSS*)
- ▶ Secure Socket Layer (*SSL*)
- ▶ Java™ applets
- ▶ Flash® images
- ▶ Characters viewable in *IE Print Preview*
- ▶ Frames

In general, anything rendered by Internet Explorer will appear in your output PDF. Set the [EngineToUse](#) property equal to 1 (*one*) to use the *IE Engine*.

NOTE: You must [Configure the WGIELogin User](#).

The *IE Engine* only includes [Properties](#) at this time.

- ▶ **Properties**

Properties

The *IE Engine* properties offer you control over many common Internet Explorer settings, such as *Authentication*, but also provide unique WebGrabber settings like wait times.

- **IEBottomMargin**
- **IEDomain**
- **IEFooter**
- **IEHeader**
- **IELeftMargin**
- **IEOrientation**
- **IEPassword**
- **IEPrintFrames**
- **IERightMargin**
- **IETopMargin**
- **IEUserName**
- **IEWaitBeforePrinting**

IEBottomMargin

Description: The size of the bottom margin for the output.

Syntax: `object.IEBottomMargin = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	FLOAT	The distance between the bottom edge of the printed page and rendered data in inches for the output PDF.

IEDomain

Description: The domain used for Internet Explorer NTLM authentication.

Syntax: `object.IEDomain = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The domain. Defaults to "." for the local domain.

IEFooter

Description: Defines page footer for the output PDF.

NOTE: The font cannot be changed.

Syntax: `object.IEFooter = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>A value passed as the page footer using the following replacement parameters:</p> <p>&w = The current dialog window title.</p> <p>&u = The URL for the current page.</p> <p>&d = The short date format specified in the <i>Windows Control Panel</i>.</p> <p>&D = The long date format specified in the <i>Windows Control Panel</i>.</p> <p>&t = The time in the regional format specified in the <i>Windows Control Panel</i>.</p> <p>&T = The time in 24-hour format.</p> <p>&p = The current page number.</p> <p>&P = The total number of spooled pages.</p> <p>&& = A single ampersand (&).</p> <p>&b = Centers the following text.</p> <p>&b&b = Right-justifies the following text.</p> <p>For example, "&w&bPage &p of &P".</p>

IEHeader

Description: Defines page header for the output PDF.

NOTE: The font cannot be changed.

Syntax: `object.IEHeader = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>A value passed as the page header using the following replacement parameters:</p> <p>&w = The current dialog window title.</p> <p>&u = The URL for the current page.</p> <p>&d = The short date format specified in the <i>Windows Control Panel</i>.</p> <p>&D = The long date format specified in the <i>Windows Control Panel</i>.</p> <p>&t = The time in the regional format specified in the <i>Windows Control Panel</i>.</p> <p>&T = The time in 24-hour format.</p> <p>&p = The current page number.</p> <p>&P = The total number of spooled pages.</p> <p>&& = A single ampersand (&).</p> <p>&b = Centers the following text.</p> <p>&b&b = Right-justifies the following text.</p> <p>For example, "&w&bPage &p of &P".</p>

IELeftMargin

Description: The size of the left margin for the output.

Syntax: `object.IELeftMargin = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	FLOAT	The distance between the left edge of the printed page and rendered data in inches for the output PDF.

IEOrientation

Description: Specifies the page orientation for the output PDF.

Syntax: `object.IEOrientation = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	SHORT	A value corresponding to the output page orientation. 0 = Portrait. 1 = Landscape.

IEPassword

Description: The password for the [IEUserName](#).

Syntax: `object.IEPasswOrd = "setVAlue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The password for the IEUserName . Defaults to blank.

IEPrintFrames

Description: Instructs WebGrabber to print the active frame only.

Syntax: `object.IEPrintFrames = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	SHORT	A value instructing WebGrabber to print the active frame. 0 = All frames are printed. <i>(Default)</i> 1 = Only the active frame is printed.

IERightMargin

Description: The size of the right margin for the output.

Syntax: `object.IERightMargin = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	FLOAT	The distance between the right edge of the printed page and rendered data in inches for the output PDF.

IETopMargin

Description: The size of the top margin for the output.

Syntax: `object.IETopMargin = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	FLOAT	The distance between the top edge of the printed page and beginning of the rendered data in inches for the output PDF.

IEUserName

Description: The user name for the specified [IEDomain](#).

NOTE: The [WGIELogin User](#) must have **Act as the Operating System** rights.

Syntax: `object.IEUserName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The user name for the IEDomain . Defaults to blank.

IEWaitBeforePrinting

Description: Specifies the time WebGrabber waits before printing the [URL](#).

NOTE: This can be useful if the [URL](#) includes images or objects loaded asynchronously to the page, which might not load before the printing begins.

Syntax: `object.IEWaitBeforePrinting = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	The page load timeout in seconds used before printing. Defaults to 0 (<i>zero</i>).

Common to Both Engines

Several key **Methods** and **Properties** are common to both the **Native** or **IE** engines. These include **methods** and **properties** required in every WebGrabber script.

- **Methods**
- **Properties**

Methods

Several of these methods are required use the [Native](#) or [IE](#) engine, including [DoPrint](#) and [Wait](#).

- [AboutBox](#)
- [CFGGetPrt2DiskSettings](#)
- [CleanUp](#)
- [ClearFooterHTML](#)
- [ClearHTMLText](#)
- [ClearHeaderHTML](#)
- [ClearVariables](#)
- [DoPrint](#)
- [GetStatus](#)
- [Wait](#)

AboutBox

Description: Instructs WebGrabber to display the *About WebGrabber Client* dialog, which provides version and copyright information.

NOTE: If used during conversion, you must click **OK** to close the dialog and continue the script.

Syntax:

```
object.AboutBox = setValue
```

Arguments:

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.

CFGetPrt2DiskSettings

Description: Retrieves the activePDF Server™ [ToString](#) settings in ColdFusion® 4.5 only. If not called, the current [Server Configuration Editor Settings](#) are used.

NOTE: Must be called prior to [DoPrint](#). For all other languages and versions, use, use the [Prt2DiskSettings](#) property.

Syntax: `"retValue" = object.CFGetPrt2DiskSettings`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

retValue:	Value Type	Description & Settings
	STRING	The activePDF Server ToString settings.

Remarks: For [JPEG output](#), you must call the activePDF Server [OutputDirectory](#) and [NewDocumentName](#) properties.

For [remote implementations](#), you must call the activePDF Server [LoadRemoteProfile](#) method prior to setting and retrieving the [ToString](#) settings.

CleanUp

Description: Cleans up all internal string variables used during conversion prior to the default limit, which can be useful under heavy load. By default, this data is deleted three minutes after the printing begins.

Syntax: `object.CleanUp "ServerIPAddress", ServerPort`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	ServerIPAddress	STRING	The server IP address (<i>Typically, 127.0.0.1</i>).
	ServerPort	LONG	The port assigned to WebGrabber (<i>By default, 64320</i>).

ClearFooterHTML

Description: Clears all calls passed to the [FooterHTML](#) and [FooterURL](#) properties.

Syntax: `object.ClearFooterHTML`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

ClearHTMLText

Description: Clears all calls passed with the [CreateFromHTMLText](#) property.

Syntax: `object.ClearHTMLText = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

ClearHeaderHTML

Description: Clears all calls passed with the [HeaderHTML](#) and [HeaderURL](#) properties.

Syntax: `object.ClearHeaderHTML`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

ClearVariables

Description: Instructs WebGrabber to clear all variables passed during conversion.

NOTE: This can be useful when a single instance of the [WebGrabber Object](#) is used repeatedly.

Syntax: `object.ClearVariables = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

DoPrint

Description: Submits the PostScript® or the rendered page printed from the engine to activePDF Server for PDF generation.

Syntax: `retValue = object.DoPrint ("ServerIPAddress", ServerPort)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	ServerIPAddress	STRING	The server IP address (<i>Typically, 127.0.0.1</i>).
	ServerPort	LONG	The port assigned to WebGrabber (<i>By default, 64320</i>).

retValue:	Value Type	Description & Settings
	LONG	A value indicating the success of the operation: -99 = The server IP address is invalid or incorrect. 0 = Success.

GetStatus

Description: Returns the status of the WebGrabber operation by polling the server-machine using the internal conversation ID.

NOTE: Not recommended unless you add *Sleep* or *Wait* component to your script to prevent overloading of the CPU. See the [Wait](#) method instead.

Syntax: `"retValue" = object.GetStatus ("ServerIPAddress", ServerPort)`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
ServerIPAddress	STRING	The server IP address (<i>Typically, 127.0.0.1</i>).
ServerPort	LONG	The port assigned to WebGrabber (<i>By default, 64320</i>).

Value Type	Description & Settings
STRING	<p>A value indicating the status of the operation:</p> <ul style="list-style-type: none"> 001 = Waiting for available thread. 002 = Connecting to URL. 003 = Engine loading HTTP text. 004 = Server is creating the printer. 005 = Engine rendering the page. 006 = Spooling the rendered page to PostScript. 008 = Established connection with available thread. 009 = Waiting for Host response from URL. 010 = Downloading embedded images. 011 = Connection to URL failed. 012 = Connection to URL cancelled. 013 = PostScript created successfully. 014 = Error printing to printer. 015 = Transmitting output via FTP.

016 = Sending output via Email protocol.

017 = FTP transmission failed.

018 = Email was unsuccessful.

019 = Success.

099 = A timeout has occurred.

Wait

Description: Returns the status of WebGrabber's operation by polling the server-machine using the internal conversation ID.

NOTE: This is done using alternate *sleeps* to prevent overloading of the CPU.

Syntax: `"retValue" = object.Wait ("ServerIPAddress", ServerPort, WaitSeconds, "AcceptedCommands")`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
ServerIPAddress	STRING	The server IP address (<i>Typically, 127.0.0.1</i>).
ServerPort	LONG	The port assigned to WebGrabber (<i>By default, 64320</i>).
WaitSeconds	INTEGER	The maximum wait time for retrieving status in seconds (<i>By default, 30</i>).
AcceptedCommands	STRING	A value corresponding to the available return codes, default is "011012014017018019". For example, if this argument is set to "002", wait returns "002" when connecting to the URL. Available commands include all <i>retValues</i> except 099.

NOTE: Specifying multiple commands may require additional **wait** calls.

retValue:	Value Type	Description & Settings
	STRING	A value indicating the status of the operation: 001 = Waiting for available thread. 002 = Connecting to URL. 003 = Engine loading HTTP text. 004 = Server is creating the printer. 005 = Engine rendering the page. 006 = Spooling the rendered page to PostScript. 008 = Established connection with available thread. 009 = Waiting for Host response from URL. 010 = Downloading embedded images.

011 = Connection to URL failed.

012 = Connection to URL cancelled.

013 = PostScript created successfully.

014 = Error printing to printer.

015 = Transmitting output via FTP.

016 = Sending output via Email protocol.

017 = FTP transmission failed.

018 = Email was unsuccessful.

019 = Success.

099 = A timeout has occurred.

Properties

Both engines support these properties, which range in function from Email protocol to the methods required for passing data to render, [CreateFromHTMLText](#) or [URL](#).

- **CreateFromHTMLText**
- **EEmailAllow8Bits**
- **EEmailCC**
- **EEmailIDFrom**
- **EEmailNameFrom**
- **EEmailReplyTo**
- **EEmailSubject**
- **EEmailText**
- **EEmailTo**
- **EngineToUse**
- **FooterFirstPageOnly**
- **FooterHTML**
- **FooterStatic**
- **FooterURL**
- **FTPDDirectory**
- **FTPHost**
- **FTPPassword**
- **FTPUserName**
- **HeaderFirstPageOnly**
- **HeaderHTML**
- **HeaderStatic**
- **HeaderURL**
- **HTMLTextToFile**
- **HTTPPostData**
- **LastError**
- **PrinterName**
- **PrinterTimeout**
- **Prt2DiskSettings**
- **SMTPServer**
- **URL**
- **UseFTP**

CreateFromHTMLText

Description: Designates raw HTML to be converted to PDF.

Syntax: `object.CreateFromHTMLText = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	Raw HTML text to convert to a PDF.

NOTE: A well-formed URL is still required. A nonexistent URL can be used. Images must contain full HTTP reference. `HTMLTextToFile` must be set to `True` when using the `IE Engine`.

Remarks: The property is additive. For example, the following code produces the same results as passing "Text hello world", which can be useful for dynamic HTML creation:

```
WG.CreateFromHTMLText="<body>"
WG. CreateFromHTMLText ="Text hello world"
WG. CreateFromHTMLText ="</body>"
```

Calling the `ClearHTMLText` method clears this property.

EEmailAllow8Bits

Description: Generates the **EmailText** for *8-bit* email transmission. The Request For Comments (*RFC*) 822 *7-bit* is used by default.

Syntax: `object.EEmailAllow8Bits = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	<p>A value instructing WebGrabber to ASCII encode the output PDF.</p> <p>1 = Encoding for extended characters (<i>7-bit</i>).</p> <p>2 = Use RFC 822 standard (<i>8-bit</i>). (<i>Default</i>)</p> <p>NOTE: <i>8-bit</i> should be used when the output contains special characters, such as foreign currency indicators. For example, the Lira.</p>

EMailCC

Description: The **Cc:** email address.

Syntax: `object.EMailCC = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>The Cc: email address. The format must be one of the following:</p> <ul style="list-style-type: none"> ▶ "User Name <user@domain.com>" ▶ "<user@domain.com>" <p>You separate multiple addresses with a comma followed by no space.</p>

EEmailIDFrom

Description: The required **From:** email address from which the output is sent.

NOTE: You must also set the **SMTPServer** and **EEmailTo** properties.

Syntax: `object.EEmailIDFrom = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The From: email address. Use the following format: "<user@domain.com>" <p>NOTE: Use the EEmailNameFrom property to prefix with a user name.</p>

E-MailNameFrom

Description: The required **From:** email address from which the output is sent.

NOTE: You must also set the **SMTPServer** and **E-MailTo** properties.

Syntax: `object.E-MailIDFrom = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The From: email address. Use the following format: "<user@domain.com>"

NOTE: Use the **E-MailNameFrom** property to prefix with a user name.

EEmailReplyTo

Description: The **Reply to:** email address.

Syntax: `object.EEmailReplyTo = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The Reply to: email address. The format must be one of the following: <ul style="list-style-type: none">▶ "User Name <user@domain.com>"▶ "<user@domain.com>"

EEmailSubject

Description: The **Subject:** of the email.

Syntax: `object.EEmailSubject = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The Subject: of the email.

EMailText

Description: The **Body** text for the email. If supported by the email client, this text precedes the embedded PDF.

Syntax: `object.EMailText = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The Body text for the email.

Remarks: If you are using complex special characters, such as those used to indicate foreign currency, use the [EMailAllow8Bits](#) property.

EMailTo

Description: A required email address where the output is sent.

NOTE: You must also set the [SMTPServer](#) and [EMailIDFrom](#) properties.

Syntax: `object.EMailTo = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>The To: email address. The format must be one of the following:</p> <ul style="list-style-type: none"> ▶ "User Name <user@domain.com>" ▶ "<user@domain.com>" <p>You separate multiple addresses with a comma followed by no space.</p>

EngineToUse

Description: Instructs WebGrabber to use the [Native Engine](#) or [IE Engine](#) for page rendering.

NOTE: Must call prior to [DoPrint](#).

Syntax: `object.EngineToUse = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	SetValue	INTEGER	A value specifying the rendering engine to use. 0 = Native Engine is used for conversions. (<i>Default</i>) 1 = IE Engine is used for conversions.

Remarks: Each engine offers advantages for conversion. Refer to the [Native Engine](#) and [IE Engine](#) information for additional details and usage requirements.

FooterFirstPageOnly

Description: Instructs WebGrabber to render the document footer on the first page only or every page.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.FooterFirstPageOnly = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	A value instructing WebGrabber to render the footer on the first page or all pages. True = The footer renders on the first page only. False = The footer renders on all pages. <i>(Default)</i>

FooterHTML

Description: Defines raw HTML passed to the output PDF as the document footer.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.FooterHTML = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>The raw HTML to pass as the document footer, which can include the following dynamic page variables:</p> <p>%cp% = A flag indicating the current page number.</p> <p>%tp% = A flag indicating the total page count.</p> <p>For example, passing <code><HTML><BODY BGCOLOR=white><CENTER><H2>Page %cp% of %tp%.</H2></CENTER></BODY></HTML></code> would print the current page of total pages, centered on white background using the <code><H2></code> tag style.</p> <p>NOTE: The full <code>http://</code> protocol must be used to pass an image in the footer. Passing a <code>vbCRLF</code> is not supported.</p>

Remarks: This property is additive. For example, the following code produces the same results as passing "Text hello world", which can be useful for dynamic footer creation:

```
WG.FooterHTML="<body>"
WG.FooterHTML="Text hello world"
WG.FooterHTML="</body>"
```

Calling `FooterURL` overwrites this property.

Calling the `ClearFooterHTML` method clears this property.

To use page variables, the `FooterStatic` property must be set equal to `False`.

FooterStatic

Description: Instructs WebGrabber to render the same document footer on every page or use dynamic page variables.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.FooterStatic = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	<p>A value instructing WebGrabber to render the footer on every page or dynamically.</p> <p>True = The footer is the same for all pages.</p> <p>False = The footer is dynamic per variables. <i>(Default)</i></p> <p>NOTE: Must be <i>False</i> to use page variables.</p>

FooterURL

Description: Defines a URL passed to the output PDF as the document footer.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.FooterURL = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The URL to pass as the footer, which can include dynamic or custom variables relating to the page. For example, <code>http://www.activepdf.com/myHeader.asp?myvar=x&totpages=x&currpage=y</code> .

Remarks: Calling **FooterHTML** overwrites this property.

Calling the **ClearFooterHTML** method clears this property.

To use variables, the **FooterStatic** property must be set equal to *False*.

FTPDirectory

Description: A valid directory for the specified **FTPHost** to place the output.

Syntax: `object.FTPDirectory = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	The directory for the FTPHost . Prefix with ("/") for an absolute reference. Defaults to relative reference.

FTPHost

Description: The FTP host used when [UseFTP](#) is enabled.

Syntax: `object.FTPHost = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The FTP host used when UseFTP is enabled.

FTPPassword

Description: The password for the specified [FTPUserName](#).

Syntax: `object.FTPPassword = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The password for the specified FTPUserName . Use empty quotes ("") for no password. Defaults to no password.

FTPUserName

Description: The user name for the specified [FTPHost](#).

Syntax: `object.FTPUserName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The user name for the FTPHost . Use empty quotes ("") for Anonymous. Defaults to Anonymous.

HeaderFirstPageOnly

Description: Instructs WebGrabber to render the document header on the first page only or every page.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.HeaderFirstPageOnly = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	A value instructing WebGrabber to render the header on the first page or all pages. True = The header renders on the first page only. False = The header renders on all pages. <i>(Default)</i>

HeaderHTML

Description: Defines raw HTML passed to the output PDF as the document header.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.HeaderHTML = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	<p>The raw HTML to pass as the document header, which can include the following dynamic page variables:</p> <p>%cp% = A flag indicating the current page number.</p> <p>%tp% = A flag indicating the total page count.</p> <p>For example, passing <code><HTML><BODY BGCOLOR=white><CENTER><H2>Page %cp% of %tp%.</H2></CENTER></BODY></HTML></code> would print the current page of total pages, centered on white background using the <code><H2></code> tag style.</p> <p>NOTE: The full <code>http://</code> protocol must be used to pass an image in the header. Passing a <code>\n</code> is not supported.</p>

Remarks: The property is additive. For example, the following code produces the same results as passing "Text hello world", which can be useful for dynamic header creation:

```
WG.HeaderHTML="<body>"
WG.HeaderHTML="Text hello world"
WG.HeaderHTML="</body>"
```

Calling `HeaderURL` overwrites this property.

Calling the `ClearHeaderHTML` method clears this property.

To use page variables, the `HeaderStatic` property must be set equal to `False`.

HeaderStatic

Description: Instructs WebGrabber to render the same document header on every page or use dynamic page variables.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.HeaderStatic = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	BOOLEAN	A value instructing WebGrabber to render the header on every page or dynamically. True = The header is the same for all pages. False = The header is dynamic per variables. (<i>Default</i>) NOTE: Must be <i>False</i> to use page variables.

HeaderURL

Description: Defines a URL passed to the output PDF as the document header.

NOTE: This property is not supported for **JPEG output**.

Syntax: `object.HeaderURL = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The URL to pass as the header, which can include dynamic or custom variables relating to the page. For example, <code>http://www.activepdf.com/myHeader.asp?myvar=x&totpages=x&currpage=y</code> .

Remarks: Calling the **ClearHeaderHTML** method clears this property.

Calling **HeaderHTML** overwrites this property.

To use variables, the **HeaderStatic** property must be set equal to *False*.

HTMLTextToFile

Description: Sends the raw data passed with [CreateFromHTMLText](#) to a temporary file during rendering, which can improve performance on large documents.

Syntax: `object.HTMLTextToFile = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	BOOLEAN	A value instructing WebGrabber to use a temporary file. True = Use temporary file. False = Do not use temporary file. <i>(Default)</i>

HTTPPostData

Description: Specifies data to be posted to the [URL](#) during conversion.

Syntax: `object.HTTPPostData = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	Data posted to the URL to convert. For example, " <i>keyword=data</i> ".

NOTE: Data must be URL encoded including items like spaces and ampersands. The *content-type* must be included for the [IE Engine](#).

LastError

Description: Retrieves an return code used by [activePDF Technical Support](#).

Syntax: `retValue = object.LastError`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>+# = A Windows® error.</p> <p>0 = Success.</p> <p>-# = An activePDF WebGrabber error.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

PrinterName

Description: The name of a physical printer to print the specified [URL](#).

NOTE: To generate a PDF, use the [Prt2DiskSettings](#) property instead.

Syntax: `object.PrinterName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The name of a physical printer.

PrinterTimeout

Description: The time allotted for the PDF to be printed to the printer created by activePDF Server®.

Syntax: `object.PrinterTimeout = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	The printing timeout in seconds. Defaults to 30.

NOTE: Larger documents might benefit from longer timeouts.

Prt2DiskSettings

Description: Accepts the activePDF Server™ [ToString](#) settings. If not called, the current [Server Configuration Editor Settings](#) are used.

NOTE: Must be called prior to [DoPrint](#). For ColdFusion® 4.5 only, use the [CFGetPrt2DiskSettings](#) method.

Syntax:

```
object.Prt2DiskSettings = "setValue"
```

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The activePDF Server ToString settings.

Remarks: For [JPEG output](#), you must call the activePDF Server [OutputDirectory](#) and [NewDocumentName](#) properties.

For [remote implementations](#), you must call the activePDF Server [LoadRemoteProfile](#) method prior to setting and retrieving the [ToString](#) settings.

SMTPServer

Description: The Simple Mail Transfer Protocol (*SMTP*) server used for the email properties.

NOTE: You must also set the [EMailTo](#) and [EMailIDFrom](#) properties.

Syntax: `object.SMTPServer = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	STRING	The STMP server used for email operations.

URL

Description: The Uniform Resource Locator (*URL*) to convert to PDF.

NOTE: Raw HTML can be passed for conversion using the [CreateFromHTMLText](#) property.

Syntax: `object.URL = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the WebGrabber object.
setValue	STRING	The URL to convert to PDF. A query string can be passed by appending a question mark ("?") to the URL.

UseFTP

Description: Uploads the output PDF to the designated [FTPHost](#).

Syntax: `object.UseFTP = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the WebGrabber object.
	setValue	INTEGER	A value instructing WebGrabber to post to the FTP. 0 = Do not use FTP. (<i>Default</i>) 1 = Use FTP.

Advanced Configuration

WebGrabber includes several advanced techniques designed to expand the possibilities of your implementation. This section details these techniques, including step by step instructions and examples.

NOTE: We strongly recommend that you familiarize yourself with the basic operations of WebGrabber before implementing advanced configurations.

- **Adding Page Breaks**
- **Using WebGrabber in .NET**
- **Remote Operation**
- **Preserving Background Colors, Images, and Table Borders**
- **Using the Registry Switches**

Using WebGrabber in .NET

The following instructions are for implementing activePDF WebGrabber in a .NET environment using the `APWebGrbNET.DLL` *COM Interop* wrapper. Using WebGrabber in .NET requires activePDF Server and the `APServerNET.DLL` wrapper.

Installing the APWebGrbNET.DLL and APServerNET.DLL Wrappers

The `APWebGrbNET.DLL` and `APServerNET.DLL` *COM Interop* wrappers are not installed with activePDF WebGrabber and Server. After installing WebGrabber and Server, you need to download the wrappers from http://www.activePDF.com/altdownloads/Products/dotNET/APWebGrbNET_SP7.zip, and then unzip the DLLs to a location accessible by your .NET scripting software.

Namespace in .NET

In .NET, the WebGrabber namespace becomes `APWebGrbNET.APWebGrabber` and the Server namespace becomes `APServerNET.APServer`.

Referencing APWebGrbNET.DLL and APServerNET.DLL

You must add a reference to `APWebGrbNET.DLL` and `APServerNET.DLL` in your .NET scripting environment. A missing or invalid reference will result in a *missing type* or *namespace* error. Use the following steps to add a reference in Visual Studio® .NET:

1. Start Visual Studio .NET.
2. In the **Solution Explorer**, right-click **Reference**, and then select **Add Reference**.
3. In the **Add Reference** dialog, click **Browse**.
4. In the **Browse** field, type the location of the `APWebGrbNET.dll`, and then click **Open**.
5. Repeat these steps to reference the `APServerNET.DLL`.

Instantiating the Object

After referencing the .NET DLLs, you must instantiate the WebGrabber and Server objects in .NET using the syntax of the script language. Instantiation instructions for VB.NET and C# are provided below.

VB.NET

1. Add the following lines to the beginning of your .NET code:

```
Imports APServerNET
Imports APWebGrbNET
```

2. Instantiate the WebGrabber and Server objects:

```
'Instantiate the Server object.  
Dim SVR As APServerNET.APServer  
SVR = New APServer  
'Instantiate the WebGrabber object.  
Dim WGR As APWebGrbNET.APWebGrabber  
WGR = New APWebGrabber
```

C#

1. Add the following lines to the beginning of your .NET code:

```
Using APServerNET;  
Using APWebGrbNET;
```

2. Instantiate the WebGrabber and Server objects:

```
//Instantiate the Server object.  
APServerNET.APServer SVR = new APServer();  
//Instantiate the WebGrabber object.  
APWebGrbNET.APWebGrabber WGR = new APWebGrabber();
```

Remote Operation

Specifying a [URL](#) to WebGrabber from a remote location requires that you install the activePDF Server™ and WebGrabber software on both machines.

NOTE: The *server-machine* where the conversion takes place must be licensed installation. All builds must match exactly.

Installing the Remote Components

1. On the *server-machine*, log on as the Local Administrator or a User account with administrative rights.
2. Download the *latest software* from <http://www.activepdf.com/downloads/serverproducts/index.cfm>.
3. Run the installation per the [standard instructions](#).
4. Click **Start > Run**, type **services.msc /s**, and then click **OK**.
5. Right-click the **Print Spooler** service, and then click **Restart**.

NOTE: Restarting the *Print Spooler* service will restart all system printing processes.

6. Click **Start > Run**, type **services.msc /s**, and then click **OK**.
7. Repeat the steps for the *remote-machine*, but do not apply a serial number.

Submitting from a Remote Location

The following configuration is necessary for remote submission:

Location	Required Installations
Server-Machine	activePDF Server WebGrabber.
Remote-Machine	activePDF Server and WebGrabber.

The **IPAddress** must be the address of the *Server-Machine* when submitting files from the remote location. Refer to the [DoPrint](#), [GetStatus](#), and [Wait](#) methods for additional details.

NOTE: Virtual IP's are not supported.

Preserving Background Colors, Images, and Table Borders

activePDF WebGrabber provides a few options for preserving the background color and table borders in your PDF output. The steps depend on the specified [EngineToUse](#):

- ▶ [Native Engine](#)
- ▶ [IE Engine](#)

Native Engine

To preserve colors with the [Native Engine](#), you must call the [PrintBackgroundColors](#) property to print no colors, only table cell colors, or all background colors. The following VBScript code example demonstrates the use of the property:

```
' Define Variables
' EngineToUse (1=IE, 0=Native)
varEngine = 0

' URL format: "http://www.activepdf.com" or "file://folder/file.htm"
varURL = "http://www.activepdf.com/support/knowledgebase/view.cfm?kb=10026"

' Timeout for various calls
varTimeout = 60

' Set activePDF Server properties
APS.OutputDirectory varPath
APS.PDFTimeout = varTimeout

' Must call before DoPrint to pass server settings to WG
WG.Prt2DiskSettings = APS.ToString()

' Set WebGrabber properties
WG.EngineToUse = varEngine
WG.URL = varURL
WG.TimeOut = varTimeout
WG.PrinterTimeout = varTimeout
WG.PrintBackgroundColors = 1
WG.TopBottomMargin = 150
WG.LeftRightMargin = 150
```

NOTE: The [Native Engine](#) prints all images.

IE Engine

To preserve background colors, images, and table borders with the [IE Engine](#), you must set the Internet Explorer *Print background colors and images* option for the [WGIELogin User](#) per the following instructions:

NOTE: By default, *Print background colors and images* is not selected. Setting this option will affect all print jobs for the [WGIELogin User](#).

1. Log in as the [WGIELogin User](#).
2. Start *Internet Explorer*.
3. Click **Tools**, point to **Internet Options**, and then click the **Advanced Tab**.

4. Scroll down to the *Printing* options and then click to select **Print background colors and images**.
 5. Click **OK**.
 6. Restart your machine.
-

Using the Registry Switches

The following registry switches can be used to control the default or enable additional WebGrabber functionality. Do not alter or create these values unless instructed by [Technical Support](#).

CAUTION: We strongly recommend that you back up the registry before making any changes. Incorrect changes to the registry may result in permanent data loss or damaged files. Make sure that you modify only the keys specified. For additional information on editing the registry, refer to your Microsoft Windows documentation before proceeding.

Main Switches

These are the main WebGrabber registry switches found in the `HKEY_LOCAL_MACHINE\SOFTWARE\activePDF\APWebGrabber` key.

String Value	Description & Settings
<code>DebugMode</code>	<p>Enables the activePDF WebGrabber debug mode.</p> <p>0 = Disabled. <i>(Default)</i></p> <p>1 = Enabled.</p>
<code>DisablePriorityBoosting</code>	<p>Disables the operating systems standard functionality to increase <code>ProcessPriority</code> level, which can decrease performance. Can be used with <code>ProcessPriority</code>.</p> <p>NOTE: Do not set <code>DisablePriorityBoosting</code> and set <code>ProcessPriority</code> to 1.</p>
<code>ProcessPriority</code>	<p>Sets the processing level for the CPU time priority assigned during printing, which can increase performance in certain environments. Use with <code>DisablePriorityBoosting</code>. Available settings are:</p> <ul style="list-style-type: none"> ▶ 0 - Normal level processing. ▶ 1 - Idle level processing. ▶ 2 - High level processing. <i>(Default)</i> <p>NOTE: Do not set <code>DisablePriorityBoosting</code> and set <code>ProcessPriority</code> to 1.</p>
<code>WriteToLogFileOnly</code>	<p>Writes the Debug diagnostics to a log file.</p> <p>0 = Disabled. <i>(Default)</i></p> <p>1 = Enabled.</p> <p>NOTE: Diagnostics display onscreen if <code>DebugMode</code> is set to 1 and <code>WriteToLogFileOnly</code> is set to 0.</p>

Default Switches

These switches found in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\A4APWEBGRABBER\Parameters` key

control two default aspects of WebGrabber functionality.

String Value	Description & Settings
MaxThreads	The maximum number of threads used during conversion. Defaults to 1000. NOTE: Limited by the machine specifications.
SocketPort	The dedicated port assigned to activePDF WebGrabber. Defaults to 64320. NOTE: The port must remain dedicated and open to WebGrabber at all times.

IP Switches

You can create subkeys for the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\A4APWEBGRABBER\Parameters\` key to permit or deny access to activePDF WebGrabber based on the *IP Address*.

Subkey Name	Description
PermittedIP	String Value designates a permissible <i>IP Address</i> .
DeniedIP	String Value designates a denied <i>IP Address</i> .
Publish\PermittedIP	String Value designates a permissible <i>IP Address</i> .
Publish\DeniedIP	String Value designates a denied <i>IP Address</i> .
Read\PermittedIP	String Value designates a permissible <i>IP Address</i> .
Read\DeniedIP	String Value designates a denied <i>IP Address</i> .

Once the keys are created:

1. Click **New > String Value**.
2. Type the desired **IP Address** to permit or deny accordingly.
3. Click **Start > Run**, type **services.msc /s**, and then click **OK**.

NOTE: You can set a socket mask by using a 0 (*zero*) in place of a number. For example, naming the **String Value** `10.1.10.0` includes `10.1.10.2`.

4. Right-click the **Print Spooler** service, and then click **Restart**.

NOTE: Restarting the *Print Spooler* service will restart all system printing processes.

5. Click **Start > Run**, type **services.msc /s**, and then click **OK**.

Appendices

Refer to the following appendices for additional data regarding WebGrabber:

- › [PDF Coordinates and Units](#)
- › [Native Engine Supported HTML Tags](#)
- › [WebGrabber 3.5.2 Updates](#)

PDF Coordinates and Units

The x-axis extends to the right from the page origin, and the y-axis extends upward from the page origin. The unit of measurement is given as *PDF Units* where 72 *PDF Units* equal 1 *inch*. This means that the default 8.5 in. x 11 in. page is equal to 612 by 792 *PDF Units*. This measurement is found by multiplying the two dimensions by 72.

Native Engine Supported HTML Tags

The **Native Engine** supports the following HTML tags and attributes.

Tag	Supported Attributes
<A>	<i>href, name, title, target</i>
<ADDRESS>	
<AREA>	<i>coords, href, nohref, shape, target, title</i>
	
<BIG>	
<BASE>	<i>href, target</i>
<BASEFONT>	<i>color, size</i>
<BGSOUND>	<i>loop, src</i>
<BLOCKQUOTE>	
<BODY>	<i>background, bgcolor, link, vlink, text</i>
 	<i>clear</i>
<CAPTION>	<i>align</i>
<CENTER>	
<CITE>	
<CODE>	
<DD>	
<DIR>	
<DL>	
<DT>	
	
	<i>color, face, size</i>
<FORM>	<i>action, method, target</i>
<FRAME>	<i>src, name, marginwidth, marginheight, scrolling, noresize, frameborder</i>
<FRAMESET>	<i>rows, cols, border, frameborder</i>
<H1>	<i>align</i>
<H2>	<i>align</i>
<H3>	<i>align</i>
<H4>	<i>align</i>
<H5>	<i>align</i>
<H6>	<i>align</i>
<HEAD>	
<HR>	<i>align, size, width</i>
<HTML>	

<I>	
	<i>align, border, dynsrc, height, hspace, ismap, loop, src, start, usemap, vspace, width</i>
<INPUT>	<i>checked, maxlength, name, size, src, type, value</i>
<ISINDEX>	<i>prompt</i>
<KBD>	
	<i>value</i>
<LINK>	
<LISTING>	
<MAP>	<i>name</i>
<MENU>	
<META>	<i>contents, http-equiv</i>
<NOBR>	
<NORAMES>	
	<i>type, start</i>
<OPTION>	<i>selected, value</i>
<P>	<i>align</i>
<PRE>	
<S>	
<SAMP>	
<SELECT>	<i>multiple, name, size</i>
<SMALL>	
	
<SUB>	
<SUP>	
<TABLE>	<i>align, bgcolor, border, cellpadding, cellspacing, height, nowrap, width</i>
<TD>	<i>align, bgcolor, colspan, height, nowrap, rowspan, valign, width</i>
<TEXTAREA>	<i>cols, name, rows</i>
<TH>	<i>align, bgcolor, colspan, height, nowrap, rowspan, valign, width</i>
<TITLE>	
<TR>	<i>align, bgcolor, nowrap, valign</i>
<TT>	
<U>	
	
<VAR>	
<WBR>	
<WxPrinter>	

NOTE: <WxPrinter> is a *Native Engine* custom tag.

WebGrabber 3.5.2 Updates

The following updates represent those resulting from the submission of an activePDF Technical Support Ticket.

- ▶ 3.5.2 SP7
- ▶ 3.5.2 SP6
- ▶ 3.5.2 SP5A
- ▶ 3.5.2 SP4A
- ▶ 3.5.2 SP4

NOTE: If received an incident number, but do not see a corresponding update, please submit a new ticket to [activePDF Technical Support](#) including the previous ticket or incident number.

3.5.2 SP7

- ▶ Fixed an issue in which the activePDF WebGrabber *IE Engine* stops working after the installation of Windows® Server 2003 SP1.

3.5.2 SP6

- ▶ Added support for Windows Server 2003.

3.5.2 SP5A

- ▶ Fixed an issue in which the WebGrabber Service does not stop after updating the *WGIELogin* user.
- ▶ Fixed an issue in which the **IEFooter** placement is altered inconsistently by the **IEBottomMargin** is set.
- ▶ Fixed an issue in which the conversion fails when using the **FooterHTML**.

3.5.2 SP4A

- ▶ Added the **LoadImages** and **LayoutHandling** properties to the *Native Engine* to help decrease large file load time.
- ▶ Added support for Internet Explorer 5.5 and 6.0, includes the addition of the *WGIELogin.EXE*.
- ▶ Added the **HeaderFirstPageOnly** and **FooterFirstPageOnly** properties.
- ▶ Enabled the **HeaderHTML** and **FooterHTML** properties to work with the *IE Engine*, which replace the **IEHeader** and **IEFooter** properties with Internet Explorer 6.0.

3.5.2 SP4

- ▶ Fixed an issue in which documents printed with the Internet Explorer 6.0® display a dark black line in the output.
- ▶ Fixed an issue in which the **HTTPPostData** property fails with the *IE Engine*.
- ▶ Fixed an issue in which a question mark appears in the header or footer when using the **HTTPPostData**, **HeaderHTML**, and **FooterHTML** together.
- ▶ Fixed an issue in which the page break settings don't get accepted.
- ▶ Fixed an issue in which the output PDF is not searchable in PDF viewers.
- ▶ Fixed an issue in which the *WPRHandler.EXE* process remains when using an invalid URL.
- ▶ Fixed an issue in which paper size cannot be selected with Internet Explorer 6.0. This does not work with a lower version.
- ▶ Fixed an issue in which the **cleanUP** method does not release memory.